Ethane Overview

Ethane is a backwards compatible approach to enforcing access controls within edge networks. The basic mechanism employed by Ethane is to perform a permission check on the first packet of every flow. Only after the permission check has succeeded is the client given access to the requested resource.

I. Ethane Operation Overview

Similar to SANE, all connectivity within an Ethane network is regulated by a central server called the Domain Controller or DC. By default, hosts that join the network cannot connect to any other network resource except for the DC and any other routes that have been explicitly authorized for all hosts (e.g. an outgoing http proxy to the Internet).

Clients must first authenticate with the DC before gaining access to further network resources. This can be done by explicitly authenticated with the DC via a common protocol such as HTTP, or by the DC placing itself on route between hosts and third-party authentication services (such as kerberos, radius or LDAP).

Default connectivity to the DC and back to clients is established using two mechanisms. Connectivity to the DC is established through a minimum spanning tree protocol in which each switch participates. This protocol is similar to Ethernet MST except that the DC is always the root. In addition, when clients send packets to the DC, the switches "learn" the path information and use it to create a path from the DC back to the client. This is illustrated with an example below.

When a client first logs into the network, it generates an authentication requests that is routed to or through the DC. The IP address of the client machine could be configured statically or dynamically via DHCP from the switch or a DHCP server that is accessible via a static route. As the first authentication packet from the client (carrying the clients's IP) traverses each switch en-route to the DC, each switch records the source IP and allocates some state for that particular host such that packets sent from the DC to that IP are returned back to the client. Switches may perform CIDR prefix aggregation of return routes.

Once a client has successfully logged in, it can request permission to access the network. Ethane was designed to be backwards compatible, and therefore it uses the first packet of each flow for connectivity request. When a client issues a packet subsequent to authentication, the first hop switch will check its "flow-table" to see if the packet matches an established flow. If not, the packet is forwarded to the DC for a permissions check. The DC uses the global policy along with any local policies enacted by the flow destination to determine if the client has permission. If permission is granted, the DC calculates a network path for the client to use (the network path may be subject to contraints such as requiring it to go through a middlebox), contacts each switch on the path to set up a "flow" and then forwards the packet to the destination. Subsequent packets from the client will match a flow in the switch’s flow table and be forwarded along the established path.

II. Setup

An Ethane network consists of a DC and one or more Ethane "switches". The switches may be multiport as are common today, or, two-port bumps in the wire that are placed between switches in existing networks. Each switch must be configured with the DC’s public key. Similarly, the DC must be configured with a key for each switch (either public or private, we don’t specify a PKI). Initial key distribution only has to be done once.

An Ethane network supports static routes and dynamic routes. The former are set up at configuration time and provide default connectivity to some set of services with optional filtering rules applied. Static routes are useful for providing connectivity that does not require per-flow permission checks. For example, routes to the network gateway and DHCP server may be set up as static routes.

All other connectivity on the network requires permission from the DC. The first packet of each flow is forwarded by the first hop switch to the DC. Connectivity to the DC is determined by the switches using a spanning tree algorithms. Return routes from the DC to the clients are created on demand as packets are forwarded from the client to the DC.

Connectivity to the DC is provided in the same manner as SANE except that, instead of using return capabilities for the return path from the DC to a source, switches generate forwarding tables as they see traffic pass to the DC. Key exchange and passing topology information to the DC is identical to how it is done in SANE.

III. Authentication

Performing a permission check requires the DC to know the location and identity of a requestor. The location is used to determine if there are any connectivity constrains from the access point used by the requestor, if the requestor is allowed to use the given access point (useful for preventing hosts from moving, e.g. 911), and to revoke access to the DC for a misbehaving sender.

The identity is obtained by requiring each user to authenticate either with the DC or a third-party authentication service such as Kerberos, AD, Radius or LDAP. If a third-party server is used, the DC must place itself on-route between itself and the authentication server. It also must be able to determine if an authentication was successful.
As clients authenticate, switches record their IP address and location (physical port). After authentication, all subsequent packets from a host’s location and source IP are assumed to come from the client. If multiple clients share the same physical port (e.g. via Ethernet switch or multi-user machine), separate VLANs can be used to differentiate between clients. Doing so requires trusting the mechanism adding the VLAN label and cannot be enforced by Ethane.

IV. NETWORK POLICY

Ethane’s policy model allows a centrally declared policy written by the network administrator, as well as user controlled access controls. The global policy affects all communication within an Ethane network. It specifies what communication is permissible (considering source/dest user, source/dest host, source/dest access point and protocol) and what, if any, constraints there are on the route.

Global policy consists of declarations of the following form:

\[
\text{[PERMIT|DISALLOW] principal-src [principal-dst] [USING [protocol] VIA constraints]}
\]

**PERMIT:** Specifies that the following rule is permitted, but by default is not enforced. In order for the DC to allow a given communication, the receiver must also specify a local rule which allows it. If not included, all communication described by the rule is allowed.

**DISALLOW:** Restricts all communication specified by the rule.

**principal:** Network principals are users, hosts, access points and services. Declarations may include compound principals. This can be used for example to bind users to machines or machines to access points. Compound principals have the form \[u-user|s-service]:h-host:a-access-pt\].

**protocol:** Any protocol known to the system.

**constraints:** A set of constraints on the route, normally in the form of middleboxes the route must traverse.

An Ethane network can be either **default off** or **default on**. If a network is configured to be default off, no communication is allowed unless a rule explicitly allows it. PERMIT rules are applied first and then DISALLOW rules. If a network is default on, then all communication is allowed unless a rule explicitly forbids it. In this configuration, DISALLOW rules are applied first and then PERMIT rules.

The global policy can also add principles to **isolated networks**. Traffic between principals in isolated networks cannot interact in any way (including sharing queues). The purpose of supporting isolated networks is to allow multiple networks with differing levels of sensitivity to share the same network components.

User declared access controls are specified over a users services. Similar to global policy, access controls may specify which users, hosts, access points can contact a given service and what protocols are permitted. A user may also specify access restrictions on their own traffic. Changes to these restrictions may require a heavy-weight authentication scheme such as a reverse turing test. User-imposed access restrictions are put in place to hinder malware from establishing outbound connections.

V. PATH SETUP

For each packet sent by an authenticated user, the first hop switch performs to actions. It first verifies that the IP address matches the physical port, if not, the packet is dropped. The switch also checks its flow table to determine whether that packet is part of an existing flow. A flow has no strict meaning in Ethane. Flow identifiers may be two-tuples (src/dst IP addresses), three-tuples (src/dst IP, server port), four-tuples (src/dst IP, src/dst Port) or other combination of headers fields. If the packet matches an existing flow, it is forwarded to the next hop switch. If a packet doesn’t match, it is assumed to be a connection setup requests and forwarded to the DC.

On receipt of a connection setup request, the DC first associates the packet with a source user, host and access point and destination user, host and access (based on the IP addresses) and then does a permission check against the network policy. If the communication is permitted, the DC generates a flow ID over the packet headers. Which headers are used for the flow ID is dependent on the protocol and the configuration of the DC. In most cases, the flow ID will be generated over the header three tuple.

The DC must also calculate the route for the flow to take. This is done and augmented with any constraints specified by the global policy (such as middleboxes). The DC communicates the flow to each switch on its path specifying the incoming flow ID, the outgoing port and optionally an outgoing flow ID for the switch to add to the header.